Cloud Publications
International Journal of Advanced Cloud Computing and Applied Research
2017, Volume 3, Issue 1, pp. 26-40
Crossref: 10.23953/cloud.ijaccar.260

Research Article **Open Access**

Reliable Multi-cloud Storage Architecture Based on Erasure Code to Improve Storage Performance and Failure Recovery

Emmy Mugisha, Gongxuan Zhang

School of Computer Science and Engineering, Dept. of Computer Science and Technology, Nanjing University of Science & Technology, Jiangsu, Nanjing, China

Publication Date: 30 May 2017

DOI: https://doi.org/10.23953/cloud.ijaccar.260

Abstract Cloud computing is a way to increase the capacity dynamically without investing in new infrastructure, training new users as well as licensing new software. The increasing popularity of cloud storage services has lead companies that handle critical content to think about using these services for their daily storage needs. Power systems records, huge Medical datasets and financial content are partial critical content that could be moved to the cloud. To this, cloud storage providers reveal challenges in achieving content availability, reliability, storage overhead and content integrity that require high efficiency on single cloud storage architecture. Cloud storage disperses content blocks over single cloud storage nodes. This study proposes Reliable Multi-Cloud Storage Architecture (RMCSA) based on Erasure Codes, that content can be dispersed to different cloud storage providers to reduce challenges which lead to Lock-In. A Multi-Cloud Control Node will manage other Control Nodes evolved in the cloud as well as service migration across sub-clouds. Currently, replication is the most applied technique which disperses content replica but standing against content loss becomes critical, hence ineffective. To assure stability in storage costs with best practice to assure content failure or loss recovery, we applied a Maximum Distance Separable such as Reed-Solomon to our RMCSA. Analysis shows that RMCSA resolves a number of storage issues and recommends Zfec library as erasure code emulator due to its accuracy.

Keywords *multi-cloud; erasure code; cloud storage vendor; control nodes; cloud client*

1. Introduction

Nowadays, the volume of content generated for use and storage is growing daily (Sakr, et al., 2011). Accumulation is revealed when the volume of content is turning a heavy load to manage on available systems. Medical record databases, power systems and financial content are partial area of daily cumulating huge content. Cloud storage providers play a significant role handling these cumulating records flexibly with cost effectiveness in comparison to rebuilding infrastructures. In response, a cloud vendor extends interesting alternatives, i.e. dynamic capacity, pervasive access and high availability using replication (Knorr & Gruman, 2013).

Cloud computing is real, and it's growing fast. In 2009, cloud services made up 5 percent of worldwide IT spending. These services are helping organizations reduce costs, enhance scalability, increase implementation speed and improve applications and business processes. But the real promise of cloud computing lies in developing new markets and services that give clients competitive advantage in rapidly changing markets.

A pay-as-you-go model was introduced for economic realm (Armbrust, et al., 2009). The on-demand concept suggests a tenant to pay when the service is available (Armbrust, et al., 2009; Buyya, et al., 2009). Cloud providers mostly look reluctant to assure services with fixed prices. In this sense, cloud customers are vulnerable to price increase and locked. Providers may go out of business due to lock-in, thus limiting organizations from adopting cloud storage (Wikipediawriters, 2012).

Apparent method to eliminate vendor lock-in dilemma is based on shifting chunks of big content to various cloud service providers. This advancement in cloud computing context is called Inter-Cloud (Bernstein, et al., 2009). In this paper the concept is referred as ''Multi-Cloud'', Multi-Cloud is simply described as a group or a combination of cloud providers which compose a cloud i.e. cloud of clouds (Grozev & Buyya, 2014; Kelly, 2016).

2. Literature Review

Some of the previous works on the subject have dwelt on the high-availability of stored content through the replication of this content on several cloud storage providers, and thus are closely related to Multi-Cloud Security Architecture (RMCSA). The idea of spreading content across different cloud storage providers is not new, though previous work wouldn't spread content chunks of each file to different storage providers.

In (Abu-Libdeh, et al., 2010), RACS, a proxy server was used as a broker to manage transactions between customers and cloud storage providers. It employs RAID5-like techniques; mainly erasure codes, to implement high-available and storage-efficient content replication on diverse clouds. Nevertheless, there is no comprehensive analysis regarding efficiency of this method in Multi-Cloud storage services.

STRATOS is another implementation of Multi-Cloud. It focuses on automatic cloud provider selection for services resource allocation running on different cloud providers (Pawluk, et al., 2012).

Another content management implementation is Flexible Robust Intelligent Elastic Data Management (FRIEDA) (Ghoshal & Ramakrishnan, 2012; Li & Qiu, 2014). To achieve various content management strategies, it separates content control and execution. This content management is robust and elastic especially for composable infrastructure environments.

In HAIL (Bowers, et al., 2009), before transforming the file *F* into a dispersed, encoded form, they halved it into distinct blocks *F*(1)*; : : : ; F*(`) and disperse these blocks across the primary servers *S*1*; : : : ; S*`. This dispersed clear text (plaintext) form of the file remains free from encoding steps. While in RMCSA, the encoding occurs at the Control Node level and disperse blocks of file *F* to all datacenters (Servers S) in the sub-Cloud. They (HAIL) then encode each block *F* (*j*) under the server code with error rate ²*c*. HAIL aggregates cryptographic protocols for proof of recoveries with erasure codes to provide a software layer to protect the integrity and availability of the stored content, even if the individual clouds are compromised by a malicious and mobile adversary. The limitation is that, it does not assure confidentiality of the stored content.

DEPSKY is a cloud-of-clouds that uses erasure codes to overcome the limitations of individual clouds by storing only a fraction, about half of the total amount of content in each cloud (Bessani, et al., 2013). The difference with comparison with this work is that, author don't employ software-based security overloads along with content distribution to data centers.

This work considers erasure coding as a method to disperse content over different cloud storage providers, as advocated by Amazon (Amazon, 2016), it provides a general architecture for applying erasure coding concept in Multi-Cloud Storage systems. This architecture gives an in-depth view about its functionality.

This paper aims at addressing the weaknesses in the distribution of content across different cloud storage systems. The proposed architecture contributes collaborative scheme across different cloud storage systems, for robust content accessibility and failure recovery based on erasure coding.

2.1. Erasure Coding Concepts

In cloud storage system, tenants store their content in the cloud and no longer possess the content locally. Thus, the correctness and availability of the content files stored on the dispersed cloud servers must be assured. One of key issues is to effectively detect any unauthorized content modification and corruption, as a result of server compromise and/or random Byzantine failures.

In recent years, erasure codes have moved to a fore content failure prevention over storage systems with dispersed different storage disks. In the storage disk, a *k*-of-*n* systematic Maximum Distance Separable (MDS) erasure (Knorr & Gruman, 2013) takes *k content blocks* and generates *n – k parity blocks m*, such that any subset of *k* blocks can restore the *k* content blocks (original content).

In addition, extra blocks are considered as linear combinations of content blocks. From here, a *Stripe* is the set of the *k* content blocks and *m* parity blocks. i.e., If *a* & *b* are content blocks *{D0, D1,…Dk-1}*, then author could generate two parity blocks *a+b* and *a−b,* {P0,P2, P3….Pm-1}.

Given a stripe consisting of the four blocks *{a, b, a+b, a−b*}, any subset of two blocks can restore *a & b* (Figure 1).

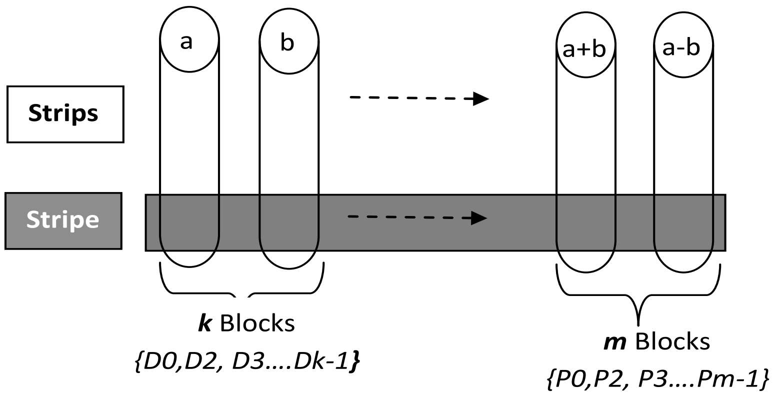


Figure 1: *Erasure coding storage system where n =4, m=2, k=2*

That is to say, *a+b & b* subsets can obtain a by subtracting b from *a+b*. Therefore, this has a 2-of-4 erasure code, which can tolerate the loss of any 2 blocks in the stripe (Figure 1). The Figure 2 below compares Erasure code versus Replication in storage systems.

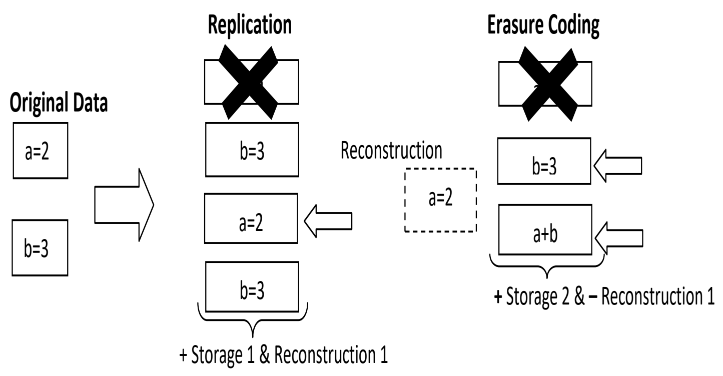


Figure 2: *Replication & erasure codes content restoration comparison*

In comparison with replication concept, this is more powerful than *2-way* replication with the same space overhead; its philosophy is as if author replicated *a* & *b*, {a, b, a, b}; thereafter, given that a failure occurred from any replica i.e. *a*, authors cannot restore *a* anymore. Furthermore, the rest theory regarding file encoding and decoding (recovery) remain in Reed Solomon code literature (Reed & Solomon, 1960).

3. Vendor Lock-in Problem

In active systems, pushing big content to cloud storage is one of the best alternatives in the science of computing. In that, cloud providers assure high availability and reliability of content. However, client/tenants or consumers may fail to access their content/services due to a number of factors, hence unable to resolve the service from anywhere.

In cloud storage system, vendor lock-in pops out when content accessibility is in vain or costly as a result of different single-based cloud architecture. Expanding deeper, assume a company stores its content (in terabytes) in cloud storage and the content is accessed concurrently by billion tenants, a tenant's access requires bandwidth (in kilobytes). With cloud charges agreements (SLA), the company will be charged storage space and bandwidth consumed. Moreover, given that the provider decides to maximize bandwidth charges, i.e. 1 cent/kilobyte. The company will face an apparent expenditure for its storage services. Therefore, the increase in expenditure will become a huge financial burden to the company to continue using this cloud storage provider. In addition to expenditure, cloud internal security management and other cloud rules may correlate similar cases where content is available but inaccessible by cloud tenant in due time.

Duplicating or replication of content is one of techniques applied to eliminate vendor lock-in problem in cloud storage systems by dispersing content blocks. Here, a company can use different cloud storage servers and disperse its content over them. Hence, sky computing as described in (Keahey & Katarzyna, 2009). Multi-cloud (or cloud of clouds) achieves service availability, elasticity and agile Service Level Agreement (SLA). In Multi-Cloud, several cloud providers are introduced and controlled independently by a client. Multi-Cloud player have no access to information from another (Grozev & Buyya, 2014).

3.1. Cloud Storage Structure

Cloud computing consist of two components, the front end and the back end. The front end of the cloud computing system comprises the client's device and applications that are needed for accessing the cloud computing system. Back end refers to the cloud itself which encompass various computers, content storage systems and servers. The whole system is administrated via a central server that is also used for monitoring client's demand and traffic ensuring smooth function of the system.

Most current Cloud storage architecture is in form of dispersed nodes from a remote database or datacenters, nodes are linked or organized inside the datacenters. Datacenters are joined together to form a storage cloud. Figure 5 presents single cloud storage architecture. The cloud storage, composed of different datacenters is controlled by a single entry point, termed as control node. This node is the main entry point for a client computer to access or gain the cloud services; it defines all storage rules, content management and technical aspects, to which cloud tenants should practice before and after requesting storage service. Many cloud storage providers are self-own-based cloud rules and technical management which creates indifferences between the different cloud storage structures in terms of service delivery.

3.2. The Issues Associated with Single Cloud Architecture

Understanding that businesses would think that single cloud based architecture could highly address their IT challenges will remain in dilemma. That is to say, each of the cloud models targets a particular business solution. IaaS highlights VM hosting that eliminates capital cost and geographical presence of the resource, but leave aside software licensing and technical support costs. The PaaS will append software licensing interests and deduct technical support expenditures. SaaS turns an application into service which erases all traditional related IT costs and tasks.

Based on the past literature, a number of issues have been set on table regarding cloud architecture design. The first on the list which is still in a theoretical stage is cloud interoperability, where customers of the single cloud decide to live in uncomfortable environment. Data security and reliability follows due to data lose and service latency issues respectively.

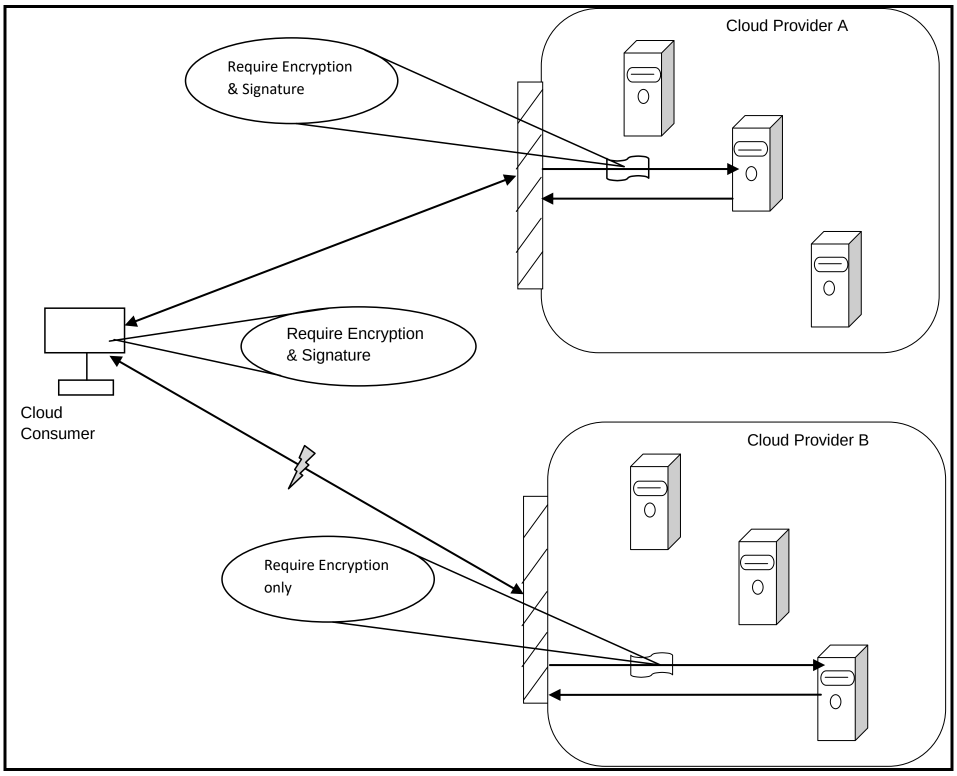


Figure 3: *Interoperability issues with regard to security policies from provider A & B*

Cloud provider' SLA (Service Level Agreement) is still independent and limited for a single cloud, it doesn't consider the framework of other providers providing identical or similar services and customers concern for their outsourced data integrity. In addition, heavy storage overheads and bandwidth consumption challenges attached to single cloud poses researchers attention to look for effective solution, such as Multi-cloud.

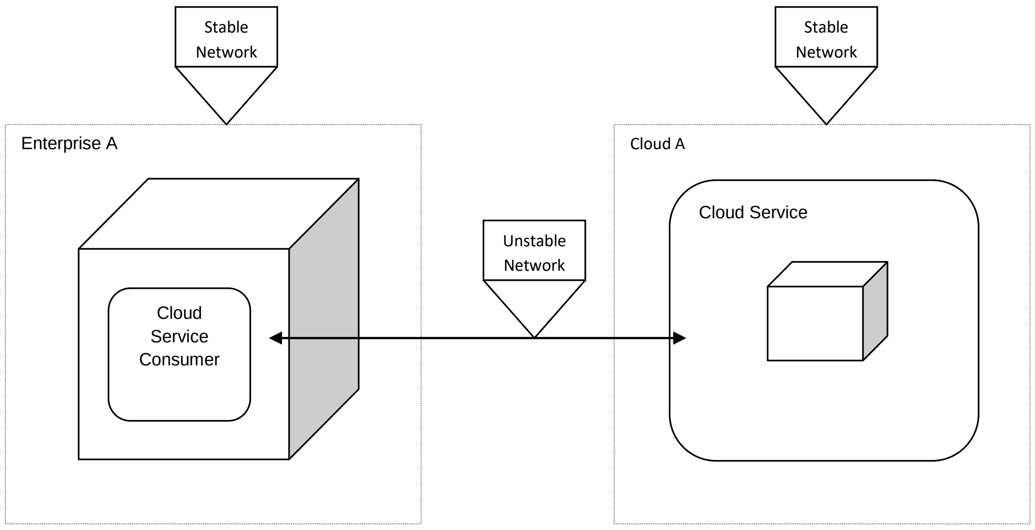


Figure 4: *Unstable network connectivity between consumer A and Cloud A*

In case a cloud storage provider takes new cloud rules based on services and accessibility load, these new rules might be inevitable to some tenants/consumers, there are no free instance migration solution flexibly currently. Instead, a tenant decides to stay rather than simply migrating to a new cloud storage provider where comparison are more efficient, hence Lock-in issues.

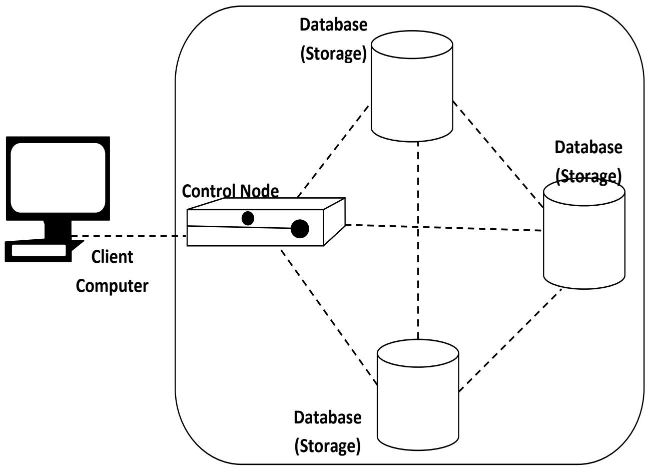


Figure 5: *Single cloud storage architecture*

4. Proposed Multi-Cloud Storage Architecture

Multi-cloud is the use of multiple cloud computing services in a single heterogeneous architecture. For example, an enterprise may concurrently use separate cloud providers for infrastructure (IaaS) and software (SaaS) services, or use multiple infrastructure (IaaS) providers. In the latter case, they may use different infrastructure providers for different workloads, deploy a single workload load balanced across multiple providers (active-active), or deploy a single workload on one provider, with a backup on another (active-passive).

There are a number of reasons for deploying a multi-cloud architecture, including reducing reliance on any single vendor, increasing flexibility through choice, and mitigating against disasters. It is similar to the use of best-of-breed applications from multiple developers on a personal computer, rather than the defaults offered by the operating system vendor. It is recognition of the fact that no one provider can be everything for everyone.

Moving from single cloud to multi-clouds is reasonable and important for many reasons. Single clouds services are still subject to outage. In addition, company may fear adversary leakages and loss of control of their content. The author assumes that the main purpose of moving to inter-clouds is to improve what was offered in single clouds by dispersing reliability, trust, and security among different cloud providers.

Every sub-cloud is a passive storage unit with five or more services or functions, for instance; List (to list files/instances/VMs), Get (reads a file), Create (creates a file/instance/VM), Put (writes or changes a file in instance/VM) and Remove (to erase a file). However, with passive storage unit, author implies that no other services than what is needed to mitigate the mentioned services or functions is executed. Here, authors assume that access control is provided by the system in order to ensure that readers are only allowed to invoke the list and get operations.

Most current cloud storage providers, their cloud architecture are built on replication technique. As a result, cloud storage accessibility become complicated; resulting cost inefficiencies based on storage capacities and bandwidth rules that do not allow tenants to reallocate their storage instances from vendor to another for better alternatives. Moreover, due to heavy storage overheads and bandwidth consumption challenges evolved in single cloud, this study introduces a new cloud storage architecture based on erasure codes to avoid storage inefficiency apparent in a number of current cloud storage providers. The Multi-Cloud concept (Figure 6) is aimed at tackling dynamic client's content management or migration across different providers, to completely unlock cloud consumers from single stylish cloud architectures.

Integrating erasure codes into Multi-Cloud systems, will improve different cloud storage systems service availability and failure recovery.

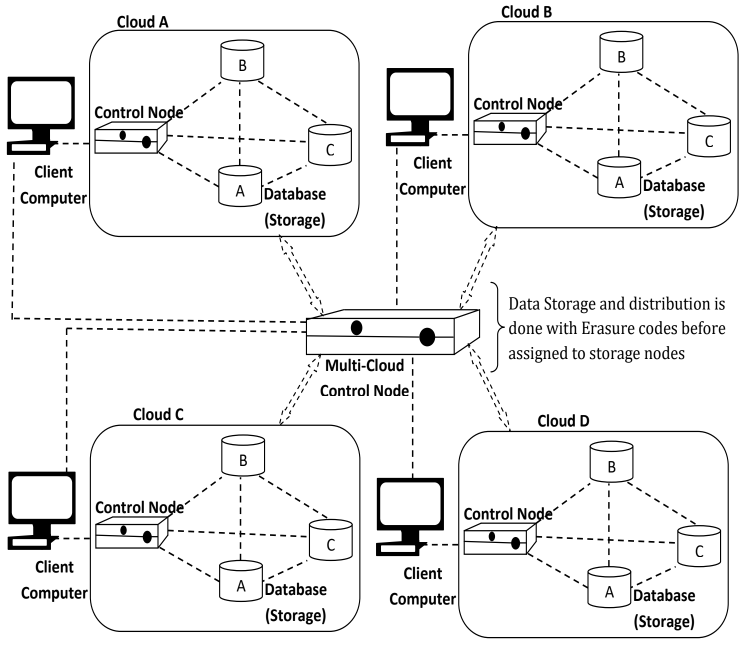


Figure 6: *Multi-Cloud storage architecture based on erasure codes*

A cloud storage tenant, having direct access session to the cloud services thought a Control Node, which provides an interface to access remote databases (datacenters) dispersed across distinct geographical locations, i.e. database storage A, B, C and more depending on cloud dispersed datacenters. These geographical remote datacenters, based on erasure codes, Control Node (CN) disperses client's file (*F*) into datacenters (D) to improve cloud storage accessibility and availability of services.

Moreover, in the process of dispersing file content, the hosting sub-cloud CN will transfer a copy of the file to a Multi-Cloud Control Node (MCN), the MCSA gate. In here, the MCN will automatically push this copy to all connected CNs (Figure 7).

```
Content Distribution Algorithm
1   Initi F as file to all Datacenters D (k+m), assign CN to MCN;
2   For i=1 to CN;
3         Copy F to all CN on MCN
4         CN_i ← F;
5               For each CN, i=1 to CN;
6                     Perform Erasure Code to F file;
7   Disperse F blocks (B)
8                        For i =1 to i=< k+m; B_i into D^th;
9                              D ← B_i;
10                       End For
11              End For
12  End For
```

These CNs specifically; excluding the hosting node (tenant base) will also automatically disperse the file content to all its datacenters based on erasure codes' concept. Thereafter, all CNs connected to MCN should agree on the same erasure codes settings while dispersing content (*F*) to their datacenters (*D*).

These CNs specifically; excluding the hosting node (tenant base) will also automatically disperse the file content to all its datacenters based on erasure codes' concept. Thereafter, all CNs connected to MCN should agree on the same erasure codes settings while dispersing content (*F*) to their datacenters (*D*).

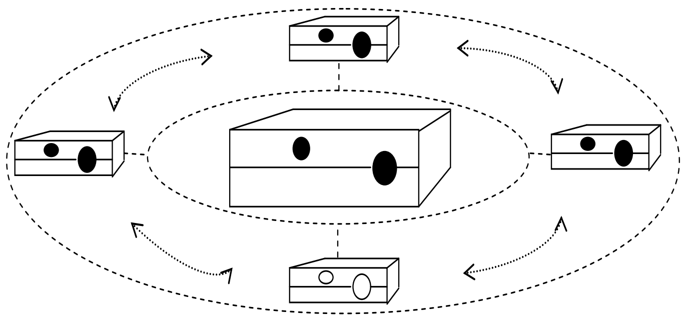


Figure 7: *Multi-Cloud control node network*

This architecture, RMCSA, based on erasure codes assure a client that given content failure or loss is detected, content can still be accessible and available from other sub-clouds on the same network flexibly through MCN.

A) Reed-Solomon (RS) Codes

This study advocates for erasure codes, Reed-Solomon is considered in RMCSA. Reed-Solomon code is characterized with Maximum Distance Separable (MDS), Redundancy and Burst Errors. From here, coding theory (Reed & Solomon, 1960) states that a stream of bits (source content) is channeled throughout communication channels, for instance telephone line. Occasionally, disruptions might occur in the channel during transmission process, causing channeled bits to mix up, e.g. 0s to 1s or 1s to 0s.

Consequently, with this Code; the strip unit is a w-bit word; where w must be huge enough that $n \leq 2^w + 1$. So that words may be manipulated efficiently, w is typically constrained so that words fall on machine word boundaries: $w \in \{8, 16, 32, \text{and } 64\}$. However, as long as $n \leq 2^w + 1$, the value of w may be chosen at the discretion of the tenant. Most implementations choose w = 8, as a result of less than 256 used disks which demonstrated best performance.

Reed Solomon codes, treat w as a number between 0 and $2^w - 1$, and it operate on these numbers with Galois Field arithmetic (GF (2^w)), which defines addition, multiplication and division on w so as to limit the system and well-responding (Macwilliams, & Sloane, 1977).

a) Encoding with Reed-Solomon

The process of encoding with Reed-Solomon codes is as simple as a linear algebra, whereby a Generator Matrix (G^T) is derived from a Vandermonde matrix, and G^T is multiplied by the k (content) to create a codeword composed of the k (content) and m (coding words). Figure 8 shows the process:

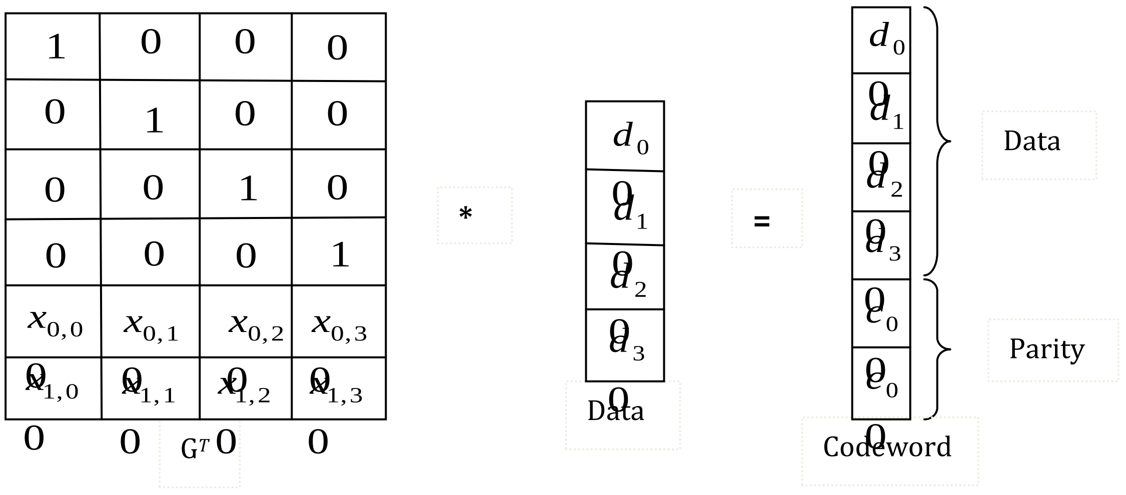


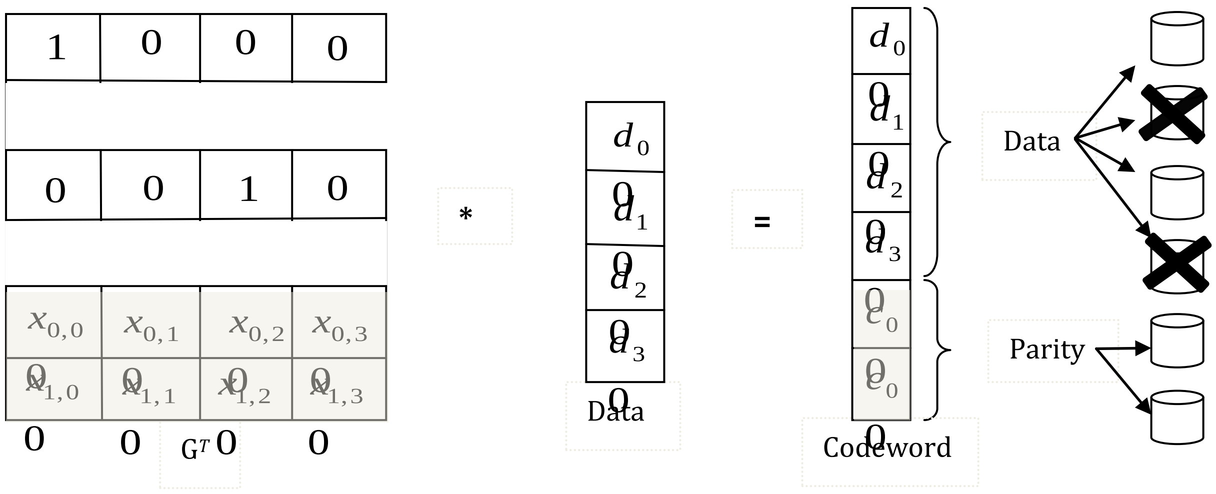
Figure 8: *Reed-Solomon coding for k = 4 and m = 2. Each element is a number between 0 and 2 w − 1.*

b) Decoding with Reed-Solomon

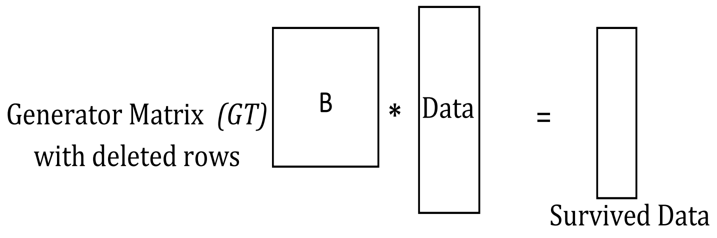
In case of disks failure, Reed-Solomon decodes by deleting affected rows of G^T, by inverting G^T, and multiplies the inverse by the survived words.

The derivation of G^T ensures that the matrix inversion is always successful.

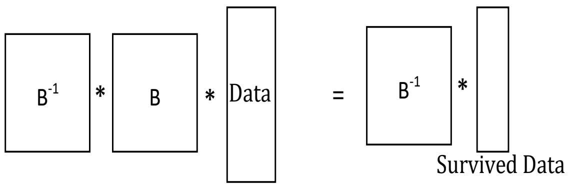
Step #1: Deleing affected rows matching to failed disks


1 0 0 0
0 0 1 0
$x_{0,0}$ $x_{0,1}$ $x_{0,2}$ $x_{0,3}$
$x_{1,0}$ $x_{1,1}$ $x_{1,2}$ $x_{1,3}$
G^T
*
d_0 d_1 d_2 d_3
Data
=
d_0 d_1 d_2 d_3 c_0 c_0
Codeword
Data
Parity

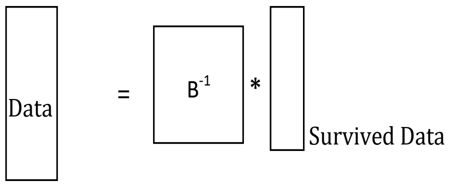
Step #2: Rewrite this equation mathematically



Step #3: Invert B and multiply it by both sides of the equation



Finally, the original content is computed out as;



In GF (2^w), addition is equivalent to bitwise exclusive-or (XOR), and multiplication is more complex, typically implemented with multiplication tables or discrete logarithm tables (Greenan, et al., 2008). Nevertheless, these codes are studied costly.

From several open-source implementations of RS coding (Plank & Schuman, 2009), this study recommends Zfec library. The Zfec, is a library for erasure coding which is built on top of a RS coding library developed for reliable multicast (Rizzo, 1997). The Zfec is based on Vandermonde matrices when w = 8. It is friendly and flexible, including a command-line tools and APIs in C, Python and Haskell.

Based on the functionality of Reed-Solomon Codes, different providers have already applied to their storage systems individually, e.g. Windows Azura Storage and the analysis show that failure recovery and content reliability demonstrated a significant change.

4.1. Architecture Effectiveness and Resources Management

In this section, authors present the proposed architectural review and other attachments related to cloud function managements. The issue of Input/Output functionality has always been at top in cloud computing, as of cloud storage systems for example. The investigations show that Input/Output hardware functionality predicts decrease than CPU execution time and memory. In addition, this issue expands as a result of the accelerating use of datacenters of PCs or physical content stores (Baker & Buyya, 1999). Consequently, Authors ague to speedup Input/Output functionality to correlate with CPU's for cloud service delivery effectively.

Generally, these systems constitute a number of IT resource pool connected by web and form a vast datacenter. These resources or nodes are the primary units of cloud datacenter. A node individually is autonomously an IT resource or virtual machine (VM). Thus, the cloud storage architecture is halved into two initial units; the application software and storage devices. The main function of application software is to implement operations on content resource and coordinate the storage devices into an efficient system.

Considering MCSA; Multi-Cloud Control Node (MCN) is serving as a cloud broker and Control Node (CN) is serving as datacenter virtual machines (VMs). More details are discussed below.

A) Multi-Cloud Control Node (MCN)

The MCN is the reception and bridge of networked cloud storage resources. MCN will receipt cloud tenant content and disperse a copy to all CNs on the cloud. If one of the networked clouds is down, tenant's content can be accessed from other remaining sub-clouds on MSCA through MCN. More importantly, all definitions of data format, migration and standardization across clouds must be defined in MCN during architectural design and implementation. This will answer the data lock-in issues lying within single cloud as a result of lacking interoperable frameworks.

B) Control Node (CN)

CN is the controller of the storage resources or nodes on MSCA. It will manage the Cloud Information Service (CIS), and the halved file blocks. The CIS executes in a virtualized mode in the cloud. CIS will play a role of cloud content and metadata management. Moreover, content blocks will be dispersed to VMs for physical content storage and access to the disks attached based on erasure code concept.

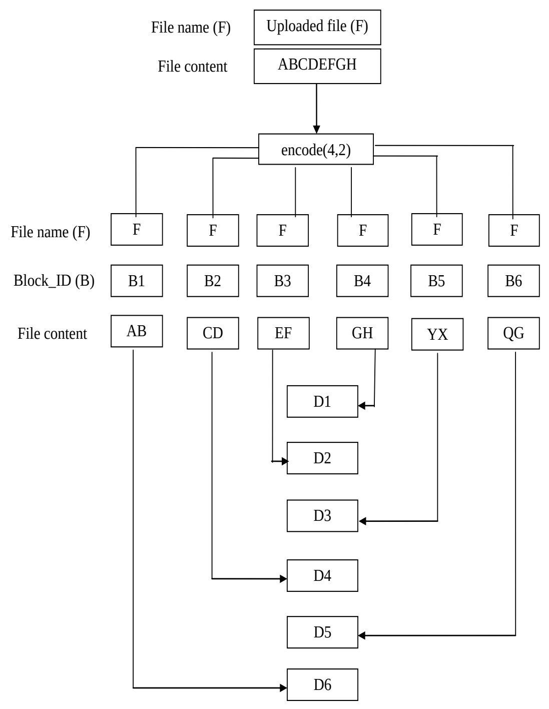


Figure 9: *CN receipted file encoded and dispersed*

The figure above elaborates important procedures that results robust storage system. The encoding process stores block files in a unique identifier (B_i) to strengthen data consistency and mutual exclusiveness during decoding process.

The VM process undergoes four processes; 1) admitting quests from the CN, 2) addressing computed file pointers to the physical file address; (3) evoking broker functions to read/write the physical content; (4) delivering back the solutions or services to the CN.

The VM addresses all Input/Output or a service bypassing the CIS, while pushing the Input/Output solutions together with success/failure notifications to the CIS. For Input/Output improved functionality, the VM must record each recent file event logs. With RMSCA, VM and CIS works hand in hand with respect to tenant' service requests through a CN. Below are some control measures on RMSCA:

Storage resource control: These include the VMs and the file content. RMSCA' services will generate operational huge amount of content/data. Improving Input/Output functionality and measurable, the CN will halve file content into blocks and disperse these blocks unto different VMs. It handles the entire system data and service scalability intercepted from MCN.

Content access: Content access services will be directed to CIS from CN. As a broker, the CIS authenticates a service or a request, and the acknowledgement or a conclusion will be sent back to the CN together with access privileges attached, if access is granted.

Metadata control: A RMSCA file logs must be captured and cached for a specific due time as metadata on the CN. During or before CN operations on file content, CN runs the log file to correlate corresponding VMs for access privileges.

C) RMSCA Workflow

Given that a tenant pushes its content *y* under rule variable *r* (as a plaintext) to the RMSCA cloud, the hosting sub-cloud CN intercepts the plaintext and send it to the MCN (RMSCA entry gate). The MCN disperses the file *y* together with *r* to all CNs on the cloud network. Every sub-cloud CN encrypts the plaintext and generates a ciphertext (*y* and r) feed back to the tenant. In case a tenant needs to access or validate a ciphertext from the cloud resource, a “key-request” is triggered to the CN. Based on the key-request string, the CN derives the rule variable *r* attached to *y*, and decide whether to issue a key or not. If it issues a key, the tenant can use it to decrypt the ciphertext computed from the RMSCA cloud.

Cryptographic Unit: In RMSCA, the CN has a cryptographic key generation with respect to tenant services. It engineers key authorization rules regarding to tenant access rule setup. Consequently, the CN provides software based security, were by trusting a service under vulnerable software based security is a MSCA weak point in terms of robust cloud security integrity, rather than hardware based security solutions. Therefore, the next future work will tackle on hardware based security details on RMSCA.

5. Conclusions

This work designed the distribution of a client’s huge content over different cloud providers to avoid lock-in and improve cloud accessibility and content failure recovery. It considers huge content as different initial file blocks. To disperse this content reliably, erasure coding concept is applied. Specifically, these initial file blocks are halved into extra blocks, and, these blocks are dispersed over certain storage nodes. Content is recoverable as long as at least a subset of blocks with minimum size exists. A block can react unobtainable for a number of factors. Given that a block is unobtainable due to corruption issues, recovery is possible. Nevertheless, given that a block is unobtainable as a result of high cost; its accessibility remains in vain. In summary, the proposed architecture shows that less number of initial file blocks increases the possibility of content unobtainable. Increasing initial number of file blocks will also improve content obtainable while failure rates become less.

This work assures accessibility and reliability of content with recovery possibility. Using Zfec library assures free implementation of erasure codes on this architecture. Therefore, this proposal will play a great improvement in the world of cloud computing hence government bodies, hospitals, financial institutions, and others join cloud adoption.

Acknowledgment

This research was partially supported by National Natural Science Foundation, under Grant NSF 61272420. The authors express their appreciation to Nanjing University of Science and Technology for creating a research fostering environment.

References

Sakr, S., Liu, A., Batista, D.M., & Alomari, M. (2011). A Survey of Huge Scale Content Management Approaches in Cloud Environments. *Communications Surveys & Tutorials, IEEE, 13*(3), 311-336.

Knorr, E., & Gruman, G. (2013). InfoWorld. http://www.infoworld.com/category/cloud-computing/

Armbrust, M., Fox, A., Griffith, R., Joseph, A.D., & Katz, R.H. (2009). Above the clouds: A Berkeley view of cloud computing. University of California, Berkeley, Tech. Rep. UCB*.* 7-13.

Buyya, R., Yeo, C.S., Venugopal, S., Broberg. J., & Brandic, I. (2009). Cloud computing and emerging IT platforms: vision, hype, and reality for delivering computing as the 5th utility. *Future Generation Computer Systems, 25*(6), 599-616.

Wikipediawriters. (2012). Vendor lock-in. *Wikipedia.*

Bernstein, David, L., Erik, S., Krishna Diamond, Steve, M., & Monique. (2009). Blueprint for the intercloud- Protocols and formats for cloud computing interoperability. In *Proceedings of the 2009 4th International Conference on Internet and Web Applications and Services, ICIW.* pp. 328-336.

Grozev, N., & Buyya, R. (2014). Inter-Cloud architectures and application brokering: Taxonomy and survey. *Software - Practice and Experience, 44*(3), 369-390.

Kelly, K. (2016). The Technium: A Cloudbook for the Cloud. http://kk.org/thetechnium/a-cloudbook-for/.

Abu-Libdeh, H., Princehouse, L., & Hakim, W. (2010). Racs- a case for cloud storage diversity. *Proceedings of the 1st ACM symposium on Cloud computing – SoCC.* p. 229.

Pawluk, P., Simmons, B., Smit, M., Litoiu, M., & Mankovski, S. (2012). Introducing STRATOS: A cloud broker service. In *Proceedings - 2012 IEEE 5th International Conference on Cloud Computing, Cloud 2012*. pp. 891–898.

Ghoshal, D., & Ramakrishnan, L. (2012). FRIEDA: Flexible robust intelligent elastic content management in cloud environments. In *Proceedings - 2012 SC Companion: High Functionality Computing, Networking Storage and Analysis, SCC 2012*. pp. 1096-1105.

Li, X., & Qiu, J. (2014). *Cloud Computing for Data-Intensive Applications* X. Li and J. Qiu (eds.) Springer.

Bowers, K.D.J., Ari, O., & Alina. (2009). HAIL : A High-Availability and Integrity Layer for Cloud Storage. *Ccs,* 489, 187-198.

Bessani, A., Correia, M., Quaresma, B., & Fernando, E.A. (2013). DEPSKY : Dependable and Secure Storage in a Cloud-of-Clouds. *ACM Transactions on Storage, 9*(4).

Amazon, Summary of the Amazon EC2 and Amazon RDS Service Disruption, 2016 http://aws.amazon.com/message/65648/

Reed, S., & Solomon, G. (1960). Polynomial Codes Over Certain Finite Fields. *Journal of the Society for Industrial and Applied Mathematics, 8*(2), 300-304.

Keahey & Katarzyna. (2009). Sky Computing. IEEE Internet Computing, pp. 43-51.

Macwilliams, F.J., & Sloane, N.J.A. (1977). The Theory of Error-Correcting Codes, Part I. Amsterdam, New York, Oxford: North-Holland Publishing Company.

Greenan, K.M., Miller, E.L., & Schwarz, T.J.E.S.J. (2008). Optimizing Galois field arithmetic for diverse processor architectures and applications. In *2008 IEEE International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems, MASCOTS*.

Plank, S.J., & Schuman, C.D. (2009). A Functionality Evaluation and Examination of Open-Source Erasure Coding Libraries For Storage. *Proceedings of the 7th Conference on File and Storage Technologies*. pp. 253-265.

Rizzo Luigi. (1997). Effective erasure codes for reliable computer communication protocols. *CM SIGCOMM Computer Communication Review*. 27(2), 24-36.

Baker Mark & Buyya, R. (1999). Cluster computing: the commodity supercomputer. *Software-Practice and Experience, 29*(6), 551-576.